

How to build Android app with Premium Mobile Code

This article will guide you to build your Android app with **Premium Mobile Code**. The article contains the following parts:

- [Prerequisite](#)
- [Set up Tools and Environment](#)
- [Set up Mobile Code](#)
- [Configure your Android app](#)

Now, let's start!



The Premium Mobile Code is ONLY available on phpFox [ULTIMATE license](#). In case you are holding other license types, please upgrade to phpFox ULTIMATE license first.

In this article, we assume that you already got the following knowledge:

- How to use Android Studio to set up and build an Android app
- How to create necessary Google certifications to build an Android app
- How to configure a Firebase project for push notification on an Android app.

Prerequisite



MUST enroll a Google Play Developer account at <https://console.developers.google.com>

Set up Tools and Environment

Install Android Studio

- Visit <https://developer.android.com/studio>, download and install the latest version of Android Studio.

Install Nodejs & Dependencies



Update

If you are using Mobile code version 1.4 or newer, you no longer need to install Nodejs & Dependencies. This step can be bypassed.

- Firstly, download and install the latest Nodejs version from <https://nodejs.org/>
- Then, install necessary dependencies, you can open the Terminal and run the following command:

```
# support eslint
npm install -g @babel/core@^7 @babel/cli@^7 eslint eslint-plugin-jest babel-eslint eslint-config-airbnb eslint-config-google eslint-plugin-flow-vars \
  eslint-plugin-import eslint-plugin-jsx-ally eslint-plugin-node eslint-plugin-promise eslint-plugin-react \
  eslint-plugin-standard eslint-plugin-mocha eslint-plugin-jasmine babel-plugin-react-intl babel-plugin-transform-remove-console \
  yarn react-native-cli create-react-native-app
```

Setup Mobile Code

- Go to Client Area and [download](#) the Premium Mobile Code (in Zip format).
- Extract Zip file to *MobileProject* folder (We just use the folder name *MobileProject* for example. You can use another folder name but should update all following commands and paths accordingly)

Install Project Dependencies



Update

If you are using Mobile code version 1.4 or newer, you no longer need to install Project Dependencies. This step can be bypassed.

Run the following command to install necessary dependencies on the project

```
#install dependencies

cd MobileProject/
yarn install
```

Configure your Android app

Configure Push Notification with Firebase

Visit Google Firebase platform (<https://console.firebase.google.com/>), create a project (You can use one Firebase project for both iOS and Android apps)

- Download *google-services.json*
- Overwrite the default file at *MobileProject/android/app/google-services.json*

Update configuration.json under folder *MobileProject/android/app/*

Default file configuration.json will be as below:

```

{
  "appName": "YourAppName",
  "bundleId": "com.yourcompany.phpfoxmobile",
  "serverUrl": "https://yourphpfoxwebsite.com",
  "clientId": "CLIENT_ID",
  "clientSecret": "CLIENT_SECRET",
  "versionName": "x.x",
  "versionBuildNumber": x,
  "facebookAppId": "xxx",
  "googlePlacesClientId": "GOOGLE_PLACES_CLIENT_ID",
  "stringKeyCodePush": "",
  "react-native-google-mobile-ads": {
    "android_app_id": "GOOGLE_ADMOB_APP_ID",
    "delay_app_measurement_init": false,
    "optimize_initialization": true,
    "optimize_ad_loading": true
  },
  "pingServerUrl": "",
  "firebaseEnabledAnalytic": true,
  "theme": {
    "splashScreenBackgroundColor": "#COLOR",
    "primaryColor": "#COLOR",
    "grayBaseColor": "#111",
    "welcomeText": {
      "fontSize": 24,
      "color": "#FFF",
      "fontWeight": 400
    },
    "welcomeImage": {
      "height": 48,
      "resizeMode": "contain"
    }
  },
  "enableAssociatedDomains": true,
  "associatedDomains": [
    "yourphpfoxwebsite.com",
    "www.yourphpfoxwebsite.com"
  ],
  "enableLiveStream": false,
  "enableRocketChat": true,
  "enableInAppPurchase": false,
  "enableAdmob": false,
  "enableAudio": false,
  "enableLog": false,
  "privateInfo": {
    "storeFile": "yourcompany.keystore",
    "storePassword": "xxxxxxxxxx",
    "keyAlias": "yourcompany",
    "keyPassword": "xxxxxxxxxx"
  }
}

```

Here is our detailed explanation for items in this file

appName

This is the app name displayed under the app logo on Android devices after installation. Ex: phpFox

bundleId

Bundle ID of your Android that you registered in Google Play Store Developer. Ex: com.yourcompany.phpfoxmobile or com.phpfox.phpfoxmobile

serverUrl

The URL of your phpFox website. Ex: <https://mobileapi.phpfox.com>

clientId and **ClientSecret**

Get these values in AdminCP > Apps > Installed > Restful API app (/admincp/app/?id=phpFox_RESTful_API)

facebookAppId

Get value in AdminCP > Apps > Installed > Facebook Login app (/admincp/setting/edit/?module-id=PHPfox_Facebook)

googlePlacesClientId

Follow instructions at <https://developers.google.com/places/android-sdk/start> to get this key for Android app.

android_app_id

Follow instructions at <https://developers.google.com/admob/android/quick-start> to get this key for Android app.

firebaseEnabledAnalytic

true OR *false*. If *true*, enable Firebase analytic. For further information take a look at [Google Analytics for Firebase](#)

versionName and **versionBuildNumber**

These two values control the versioning of your Android app in Google Play.

- *versionName* should follow the format "MAJOR.MINOR.PATCH". Ex: 1.0.3. For further information, please refer to <https://semver.org/>
- *versionBuildNumber* must be an Integer number.

primaryColor and **splashScreenBackgroundColor**

The primary color (in Hex color code, Ex: #2681D5) of your app. This color is applied to links, like button, navigation title, etc.

grayBaseColor

This color code is used for generating appropriate text color.

privateInfo

These items will be used when distributing Android app to Google Play. They can be generated within Android Studio.

Run Nodejs



Update

If you are using Mobile code version 1.4 or newer, you no longer need to run Nodejs. This step can be bypassed.

- Open Terminal, go to *MobileProject* folder
- Run the following command to start

```
node ./server.js
```

Run your Android app in Android Studio

- Start Android Studio and open Android project at *MobileProject/android* folder.



When opening, you may receive some notice messages about missing libraries of Android Studio with instructions to install them. You can follow instructions to install missing libraries.

- Now you can start running your Android app in the simulator in Android Studio